Performance Analysis Report

Post-Quantum Cryptography vs RSA-2048

Test Environment:

- System: Ubuntu 24.04 LTS
- Network: LAN
- Measurement Method: Wireshark packet capture over 10+ runs
- Algorithms Used: Kyber-768 (ML-KEM), Dilithium3 (ML-DSA-65)

1. KEY AND SIGNATURE SIZE COMPARISON

Component	Post-Quantum	RSA-2048	Size Increase
Key Exchange			
Public Key	1184 bytes (Kyber-768)	256 bytes	4.6× larger
Ciphertext	1088 bytes (Kyber-768)	256 bytes	4.2× larger
Digital Signatures			
Public Key	1952 bytes (Dilithium3)	256 bytes	7.6× larger
Signature	3309 bytes (Dilithium3)	256 bytes	12.9× larger
Total Handshake Payload			
Crypto Material	~5.9 KB (Kyber + Dilithium)	~0.8 KB	9.4× larger

Analysis: Post-quantum algorithms require significantly more bandwidth due to lattice-based constructions using Module-LWE. However, modern browser benchmarks show that 1.5 KB is negligible on 2025 networks where even mobile connections can handle this overhead without impact.

2. SECURITY COMPARISON

Algorithm	Type	Quantum Vulnerable	Classical Security	Quantum Security	NIST Status
RSA-2048	Public-key KEM & Signature	Yes (Shor)	112-bit	0-bit	Legacy (deprecated 2030)
Kyber-768	KEM	Resistant	192-bit	192-bit	NIST Level 3 (Standardized 2024)
Dilithium3	Signature	Resistant	192-bit	192-bit	NIST Level 3 (Standardized 2024)

Key Security Insights:

- RSA-2048 is completely broken by Shor's algorithm on quantum computers.
- Kyber-768 provides 192-bit quantum security, exceeding NIST security level 3 requirements.
- Dilithium3 provides 192-bit quantum security, suitable for most applications requiring long-term security.
- Post-quantum algorithms rely on Module-LWE and Module-SIS mathematical problems, which have no known efficient quantum algorithms.

3. CRYPTOGRAPHIC PRIMITIVE PERFORMANCE (Milliseconds)

Operation	Kyber-768	Dilithium3	RSA-2048	PQ vs RSA
Key Generation	0.011 ms	0.022 ms	100 ms	~9000× faster
Encapsulation	0.011 ms	—	90 ms	~8200× faster
Decapsulation	0.012 ms	—	90 ms	~7500× faster
Sign	—	0.077 ms	~1.0 ms	~13× faster
Verify	—	0.028 ms	~0.045 ms	~1.6× faster
Total (all ops)	0.034 ms	0.105 ms	~281 ms	~2000× faster

Performance Summary:

Kyber-768 is dramatically faster than RSA-2048:

- Key exchange operations: ~8000× faster with AVX2 optimization
- Total operation time: 0.034 ms vs 280 ms

Dilithium3 outperforms RSA-2048:

- Signing: 13× faster (0.077 ms vs 1.0 ms)
- Verification: 1.6× faster (0.028 ms vs 0.045 ms)
- Combined operations: 10× faster overall

4. OUR HANDSHAKE PROTOCOL PERFORMANCE

Metric	New User	Existing User	Notes
Average Time	740.7 ms	721.2 ms	End-to-end (10+ runs)
Total Bytes	~7,700 bytes	~5,745 bytes	Including TCP/IP headers
Packet Count	40-43 packets	34-36 packets	Bidirectional count
Crypto Payload	~5.9 KB	~4.8 KB	PQ keys + signatures
Network Overhead	~1.8 KB	~0.9 KB	TCP/IP + protocol headers

Wireshark Images:

- New User -> new_user.jpeg
- Existing user -> new_user.jpeg

Handshake Time Breakdown

Component	Estimated Time	Percentage
Pure Crypto Operations	<1 ms	0.13%
TCP 3-way Handshake	~0.2 ms	0.03%
Network buffer	~200-300 ms	27-40%
File I/O (Dilithium keys)	~50-100 ms	7-13%
Application Processing	~340-490 ms	46-66%
Total	**740.7 ms**	**100%**

5. COMPARISON WITH STANDARD RSA-2048 TLS

Protocol	Crypto	Handshake Time	Total Bytes	Quantum Security
TLS 1.3 (RSA-2048)	Classical	100-150 ms	~1.8 KB	Vulnerable
Our Protocol (PQ)	Post-Quantum	740.7 ms	~7.7 KB	Secure

Analysis: The actual post-quantum cryptographic operations take less than 1 millisecond (0.139 ms total: 0.034 ms Kyber + 0.105 ms Dilithium). The 740 ms handshake time is dominated by:

- 8+ message exchanges vs TLS 1.3's optimized 1-RTT design
- Separate Dilithium key exchange (new users) adds 1-2 RTTs
- Explicit mutual HMAC verification (both directions) adds 2 RTTs
- File I/O operations for persistent key storage during handshake
- No session resumption or optimization yet

6. PRACTICAL OVERHEAD ANALYSIS

Bandwidth Impact:

Scenario	RSA-2048 TLS	Our PQ Protocol	Overhead
Handshake (one-time)	1.8 KB	7.7 KB	+5.9 KB

Analysis: The 5.9 KB overhead equals 0.03 seconds of 1080p video – negligible for real-time communication.

Latency Impact:

Component	Our Implementation	Optimized PQ-TLS	RSA-2048 TLS
Crypto Operations	<1 ms	<1 ms	~281 ms
Protocol Overhead	~740 ms	~70 ms	~100 ms
Total	~740 ms	~70 ms	~100 ms

7. POTENTIAL OPTIMIZATIONS

- **Reduce round trips**: Combine messages (username + Dilithium key in one packet)
- **Implement 1-RTT design**: Pre-share or cache Dilithium keys
- **Remove file I/O from critical path**: Load keys once at startup
- **Optimize message framing**: Batch type + length + data into single send()
- **Add session resumption**: Skip full handshake for repeat connections

8. REFERENCES

- https://openquantumsafe.org/liboqs/algorithms/sig/ml-dsa.html
- https://openquantumsafe.org/liboqs/algorithms/kem/ml-kem.html
- https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.204.pdf
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10937704
- https://pq-crystals.org/dilithium/
- https://pq-crystals.org/kyber/

By:

Vikas Prajapati (2022ucs0113)

Shubham Gupta (2022ucs0110)

Shivani Consul (2022ucs0109)